REMARKS

Claims 25, 27-31, 34-38, 40, 41 and 43-47 remain pending in this application. Applicant respectfully requests reconsideration and allowance of the pending claims in view of the following remarks.

I. 35 U.S.C § 102(b) Rejections

Claims 25, 27-31, 34-38, 40, 41 and 43-47 stand rejected under 35 U.S.C § 102(b), the Examiner contending that these claims are anticipated by USPN 6,263,487 to Stripf (Stripf).

a. Independent claim 25 (and all claims dependent therefrom)

Per MPEP 2131, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987) (emphasis added). In the present application, Applicant respectfully maintains that Claim 25 is not anticipated by Stripf for at least the reason that Stripf does not expressly or inherently describe "a machine-independent program in the form of a hierarchical tree." In the November 17, 2009 Office Action, the Examiner referred to p. 11 of the online Wikipedia document provided along with the April 15, 2009 Office Action for its alleged disclosure of java byte code data structure as trees.

Applicant maintains that Java byte code is merely an intermediate representation of a Java program, which is generated by the Java compiler (see again Java bytecode by Peter Haggar submitted as evidence previously in this application). Each byte code is one byte in length and represents a machine independent instruction. At runtime, the byte code is typically converted to machine dependent code via a Java Virtual Machine (JVM). Java is merely a type of language - specifically, a class-based, object-oriented programming language - and an intermediate representation of a program in byte code of this language represents an instruction, but does not expressly or inherently describe a machine-independent program in the form of a hierarchical tree.

The Examiner confirms the Applicant's position on p. 4 of the November 17, 2009 Office Action in arguing that Stripf anticipates claim 25. Specifically, the Examiner states:

> Converting the modules and functions of the structured program into objects (e.g., translates the control program into an **object-oriented machine language, e.g., java byte code**).

Accordingly, the Examiner acknowledges that the java byte code is a language. The cited Wikipedia document also makes this clear. The Wikipedia document, cited to by the Examiner, refers to "Java libraries," which may be "the compiled byte codes of source code…to support application development in Java. Examples of such libraries include… collection libraries that implement data structures, such as lists, dictionaries, trees, and sets." (emphasis added). Critically, it is the Java library that may be in the form of a tree, not the byte code. While the Java byte code (language) referred to by Stripf may be utilized to compile a Java library as evidenced by the Wikipedia document, nothing in Stripf or the Wikipedia document discloses the intermediate byte code is in the form of a hierarchical tree. Further action is required to compile the byte code of Stripf into the form of a tree. No such step to transform byte code into the form of a hierarchical tree is disclosed in Stripf.

In view of the above, Applicant maintains that Stripf does not expressly or inherently describe "a machine-independent program in the form of a hierarchical tree." Accordingly, independent claim 25 is in condition for allowance. Furthermore, claims 27- 31, 34-37 are also patentable at least based on their dependency to claim 25, as well as based on their own merits.

b. Independent claim 38 (and all claims dependent therefrom)

With respect to independent claim 38, independent claim 38 requires "a component to load the machine-independent program in the form of the at least one hierarchical tree into the corresponding components of the automation system." Applicant respectfully maintains, for at least the reasons set forth above with respect to independent claim 25, independent claim 38 is also in condition for allowance as Stripf does not expressly or inherently describe "a machine-independent program in the form of a hierarchical tree." Furthermore, claims 40-41 and 43-46 depend on claim 38 and are patentable at least based on their dependency to claim 38 as well as based on their own merits.

c. Independent claim 47

With respect to independent claim 47, independent claim 47 requires a computer program implementing a method for executing a program for an industrial automation system, comprising..."modules and functions being structured and networked using the input aids and optionally the display device so as to form a hierarchical tree as a machine-independent program." Applicant respectfully maintains, for at least the reasons set forth above with respect to independent claim 25, independent claim 47 is also in condition for allowance as Stripf does not expressly or inherently describe "a hierarchical tree as a machine independent program."

d. Dependent claims 37 and 46

In addition, Applicant maintains that dependent claims 37 and 46 provide further reasons for allowance. Claims 37 and 46 each require that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects.

As set forth above, per MPEP 2131, "[a] claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). In addition, per MPEP 2112, "[i]n relying upon the theory of inherency, the examiner must provide a basis in fact and/or technical reasoning to reasonably support the determination that the allegedly inherent characteristic necessarily flows from the teachings of the applied prior art." *Ex parte Levy*, 17 USPQ2d 1461, 1464 (Bd. Pat. App. & Inter. 1990)." (emphasis added). Applicant respectfully maintains that Stripf fails to expressly or inherently describe that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions that access the objects via containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects.

At col. 4, lines 15-21, Stripf specifically discloses: "A class of software function blocks and a class of input/output modules are deposited in bootstrap unit Bos. These classes are created by a user, for example, on a programming unit according to the requirement of a control objective to be achieved and are transmitted to a programmable controller, for example, or to a field unit." (emphasis added). The Examiner then points to a Wikipedia document that describes a container as "a class, data structure, or an abstract data type (ADT) whose instances are collections of other objects" and concludes that Stripf anticipates claims 37 and 46.

First, Stripf clearly does not expressly recite the limitations of claims 37 and 46. Second, the above-referenced disclosures fail to inherently recite the limitations of claims 37 and 46. In particular, Stripf's disclosure that a class of software function blocks is deposited in a bootstrap unit does not necessarily (inherently) require that the objects of the machine-independent program present as a hierarchical object or operator tree are assigned a collection of infrastructure services or infrastructure functions. In addition, the teachings of Stripf (and the Wikipedia document) do not necessarily (inherently) require containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects. For example, even if Stripf discloses all that the Examiner contends it discloses, Stripf could certainly require that an infrastructure service or an infrastructure function can be used only by select (and not all) objects. Further, while a container may be a class, not all classes necessarily include containers. Thus, even if Stripf discloses a class of objects, these objects do not necessarily include containers. In view of the above, Stripf does inherently describe "containers assigned to the objects such that an infrastructure service or an infrastructure function can be used by all the objects" as claimed in dependent claims 37 and 46, nor does Stripf expressly describe the subject claim limitations. Accordingly, dependent claims 37 and 46 provide further reasons for allowance.

II. Conclusion

Applicant respectfully requests that the Examiner reconsider the rejections and timely pass the application to allowance. Please grant any extensions of time required to enter this paper. The commissioner is hereby authorized to charge any appropriate fees due in connection with this paper, including fees for additional claims and terminal disclaimer fees, or credit any overpayments to Deposit Account No. 19-2179.

Respectfully submitted,

Dated: Dec. 28, 2009

By: Janet D Hood
Janet D. Hood
Registration No. 61,142
(407) 736-4234

Siemens Corporation
Intellectual Property Department
170 Wood Avenue South
Iselin, New Jersey 08830